



# **EyeCandy3D User Manual**

A 3D Scenegraph built on OpenGL

EVELIN Project



# Inhaltsverzeichnis

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>First steps</b>	<b>2</b>
2.1	Creating a window . . . . .	2
2.2	Creating a scene . . . . .	2
2.3	Resource registry . . . . .	2
<b>3</b>	<b>Input</b>	<b>3</b>
3.1	Input events . . . . .	3
3.2	EventSystem . . . . .	3
<b>4</b>	<b>GUI</b>	<b>4</b>

## Introduction

# 1

---

EyeCandy3D is a scene graph based on OpenGL. Every program consists of one **Application**, which holds a number of windows. Each window can hold a variable number of scenes, which can run simultaneously.

## First steps

---

# 2

### 2.1 Creating a window

Custom windows must be derived from **ec::Window**. This window has to be with the window creation function provided by the application.

### 2.2 Creating a scene

Custom scenes must be derived from **ec::Scene**. By registering it with the scene manager in the associated window, it can receive updates.

### 2.3 Resource registry

Built in types:

-

# Input

# 3

---

Input is provided through the GLFW library.

## 3.1 Input events

Input events are always bound to one window. There are multiple sources, which can generate input events:

- **Mouse**
- **Keyboard**
- **Window**
- **Joystick**

An **InputEvent** consists of an **InputType** and the **EventData**. The **EventData** holds all different kinds of events, of which only one can be active at a time due to it being a union. The **InputType** describes, which part of the **EventData** is active. All other elements inside **EventData** are invalid!

I.e. if the **InputType** is *key\_pressed*, only the **KeyboardEvent** inside the **EventData** is active.

## 3.2 EventSystem

An event system is always linked to exactly one window.

### **DeviceRegistry:**

Contains input devices (mouse, keyboard, joystick etc.). Those devices can be activated to generate input events.

### **InputObservable:**

**InputListener** can be registered at this component, which will then be notified about incoming **InputEvents**.

### **InputListener:**

Input listener contain a number of callbacks, which can be added by the user. Input listener have to be registered at an **InputObservable** located in a window, to be informed about input events.

## GUI

# 4

---

The GUI is built using the Agui-library with custom backends for OpenGL.

**Abbildungsverzeichnis**

**Tabellenverzeichnis**